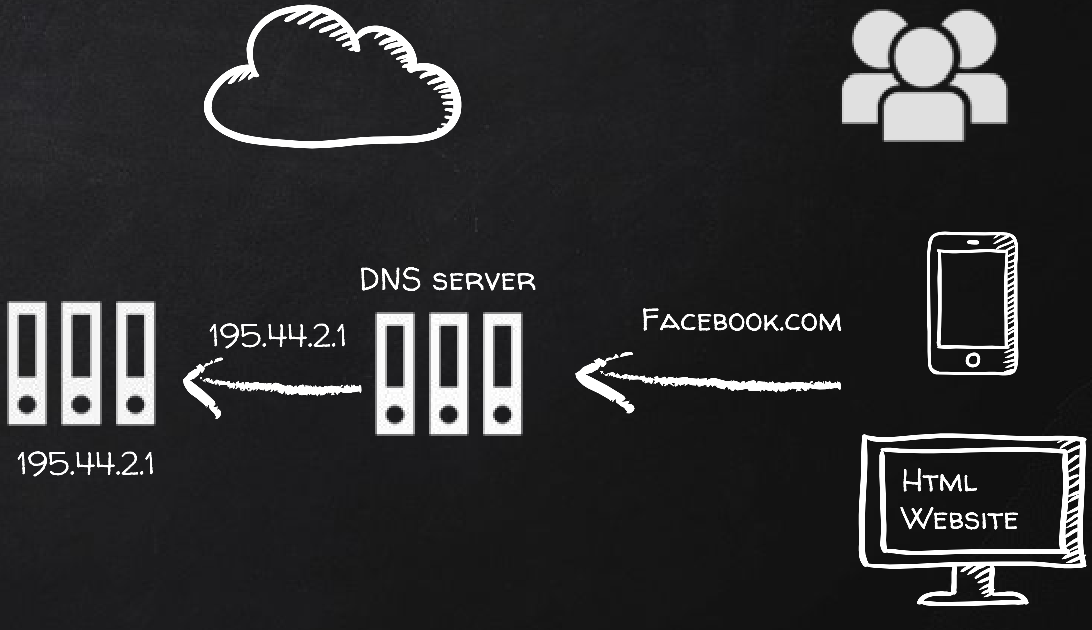
Web Penetration Testing

What is a Website

How to hack a Website?

- Computer with OS and some servers.
- Apache, MySQL ...etc
- Contains web application.
- PHP, Python ...etc
- Web application is executed here and not on the client's machine



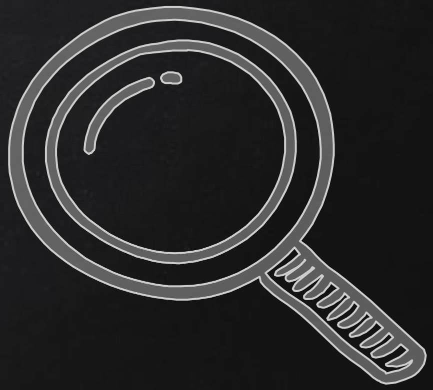
What is a Website

How to hack a Website?

- An application installed on a computer . → web application pentesting
- Computer uses an OS + other applications → server side attacks.
- Managed by humans → client side attacks.

Information Gathering

- IP address.
- Domain name info.
- Technologies used.
- Other websites on the same server.
- DNS records.
- Unlisted files, sub-domains, directories.



Information Gathering

1. Whois Lookup – Find info about the owner of the target.
   → http://whois.domaintools.com/
2. Netcraft Site Report – Shows technologies used on the target.
   → http://toolbar.netcraft.com/site_report?url=
3. Robtex DNS lookup – Shows comprehensive info about the target website.
   → https://www.robtex.com/

Information Gathering

Websites on the same server

- One server can serve a number of websites.
- Gaining access to one can help gaining access to others.

To find websites on the same server:

1. Use Robtex DNS lookup under “names pointing to same IP”.
2. Using bing.com, search for ip: [target ip]

Information Gathering

Subdomains

- Subdomain.target.com
- Ex: beta.facebook.com

Knock can be used to find subdomains of target

1. Download it > git clone https://github.com/guelfoweb/knock.git
2. Navigate to knock.py. > ce knock/knock.py
3. Run it > python knock.py [target]

Information Gathering

Files + Directories

- Find files & directories in target website
- A tool called drib.

```
> dirb [target] [wordlist] [options]
```

For more info run

```
> man dirb
```

Exploitation

File Upload Vulns

- Simples type of vulnerabilities.
- Allow users to upload executable files such as php.

Upload a php shell or backdoor, ex: weevly

1. Generate backdoor > weevly generate [passord] [file name]
2. Upload generated file.
3. Connect to it > weevly [url to file] [password]
4. Find out how to use weevly > help

Exploitation

Code Execution Vulns

- Allows an attacker to execute OS commands.
- Windows or linux commands.
- Can be used to get a reverse shell.
- Or upload any file using wget command.
- Code execution commands attached in the resources.

Exploitation

Local File Inclusion

- Allows an attacker read ANY file on the same server.
- Access files outside www directory.

Exploitation

Remote File Inclusion

- Similar to local file inclusion.
- But allows an attacker read ANY file from ANY server.
- Execute php files from other servers on the current server.
- Store php files on other servers as .txt

Mitigation

1. File Upload Vulns – Only allow safe files to be uploaded.

2. Code Execution Vulns:
   - Don't use dangerous functions.
   - Filter use input before execution.

3. File inclusion:
   - Disable allow_url_fopen & allow_url_include.
   - Use static file inclusion.

Exploitation - SQL Injection

What SQL ?

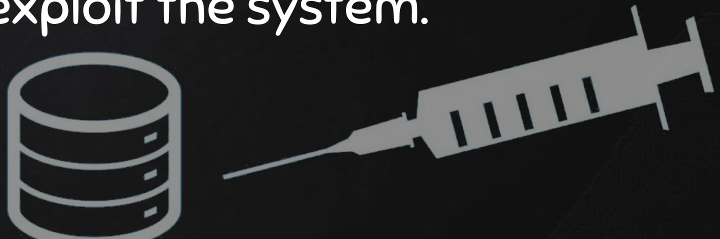
- Most websites use a database to store data.
- Most data stored in it (usernames, passwords ..etc)
- Web application reads, updates and inserts data in the database.
- Interaction with DB done using SQL.



Exploitation – SQL Injection

Why are they so dangerous

1. They are everywhere.
2. Give access to the database → sensitive data.
3. Can be used to read local files outside www root.
4. Can be used to log in as admin and further exploit the system.
5. Can be used to upload files.



Exploitation – SQL Injection

Discovering SQLi

- Try to break the page.
- Using ‘and’, ‘order by’ or “ ‘ ”.
- Test text boxes and url parameters on the form
  http://target.com/page.php?something=something



Exploitation – SQL Injection

SQLmap

- Tool designed to exploit sql injections.
- Works with many db types, mysql, mssql ...etc.
- Can be used to perform everything we learned and more!

```
> sqlmap --help
> sqlmap -u [target url]
```



Preventing SQLi

- Filters can be bypassed.
- Use black list of commands? Still can be bypassed
- Use whitelist? Same issue

→ Use parameterized statements, separate data from sql code.

Exploitation - XSS Vulns

XSS - Cross Site Scripting vulns

- Allow an attacker to inject javascript code into the page.
- Code is executed when the page loads.
- Code is executed on the client machine not the server.

Three main types:

1. Persistent/Stored XSS
2. Reflected XSS
3. DOM based XSS



Exploitation – XSS Vulns

Discovering XSS

- Try to inject javasript code into the pages.
- Test text boxes and url parameters on the form
  http://target.com/page.php?something=something

XSS
Cross Site Scripting

Exploitation – XSS Vulns

Reflected XSS

- None persistent, not stored.
- Only work if the target visits a specially crafted URL
- EX
  http://target.com/page.php?something=<script>alert(“XSS”)</script>

XSS
Cross Site Scripting

Exploitation – XSS Vulns

Stored XSS

- Persistent, stored on the page or DB.
- The injected code is executed everytime the page is loaded.

XSS

Cross Site Scripting

Exploitation - XSS Vulns

Exploiting XSS

- Run any javascript code.
- Beef framework can be used to hook targets.
- Inject Beef hook in vulnerable pages.
- Execute code from beef.

XSS

Cross Site Scripting

Preventing XSS Vulns

- Minimize the usage of user input on html.
- Escape any untrusted input before inserting it into the page.

Char	Result
&	→ &
<	→ <
>	→ >
"	→ "
'	→ '
/	→ /

→https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet

Zed Attack Proxy ZAP

- Automatically find vulnerabilities in web applications.
- Free and easy to use.
- Can also be used for manual testing.